

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
1	BRS	L1	172	reconfigurat\$5 same processor same register	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:18	
2	BRS	L2	41	1 and vector	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:04	
3	BRS	L3	3	2 and displacement	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:05	
4	BRS	L4	4	2 and compiler	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:07	
5	BRS	L5	2	4 not 3	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 13:59	
6	BRS	L6	47	1 and (generat\$5 with instruction)	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:00	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
7	BRS	L7	16	6 and vector	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:00	
8	BRS	L9	0	8 not 3	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:00	
9	BRS	L8	3	7 and displacement	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:01	
10	BRS	L10	4	instruction same load same reconfigurat\$5 same processor same register	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:04	
11	BRS	L11	2	10 not 8	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:03	
12	BRS	L12	54	instruction same load same reconfigurat\$5 and processor and register	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:04	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
13	BRS	L13	50	12 not 10	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:04	
14	BRS	L14	26	13 and vector	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:04	
15	BRS	L15	1	14 and displacement	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:05	
16	BRS	L16	21	14 and compiler	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:07	
17	BRS	L17	21	16 and operand	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:08	
18	BRS	L18	21	17 and program	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:11	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
19	BRS	L19	0	17 and second near5 configuration	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:10	
20	BRS	L20	0	17 and second with configuration	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:10	
21	BRS	L21	0	17 and executin adj unit	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:11	
22	BRS	L22	0	17 and execution adj unit	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:12	
23	BRS	L23	21	17 and execution	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:12	
24	BRS	L24	0	23 and scalar near3 data	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:13	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
25	BRS	L25	21	23 and scalar	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:18	
26	IS&R	L26	1159	(712/229,24,2,10,15,226).CCLS.	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:18	
27	BRS	L27	15	26 and reconfigurat\$5 same processor same register	US- PGPUB; USPAT; EPO; JPO; DERWEN T; IBM_TDB	2004/11/12 14:19	



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

instruction and load and reconfigurat\$5 and processor and reg



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

instruction and load and reconfigurat\$5 and processor and register

Found 30,046 of 145,831

 Sort results
by

 Display
results

☒ [Save results to a Binder](#)
☒ [Search Tips](#)
☐ [Open results in a new window](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Checkpoint Processing and Recovery: Towards Scalable Large Instruction Window Processors](#)

Haitham Akkary, Ravi Rajwar, Srikanth T. Srinivasan

 December 2003 **Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture**
Full text available: [pdf\(419.08 KB\)](#)
[Publisher Site](#)

 Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Large instruction window processors achieve high performance by exposing large amounts of instruction level parallelism. However, accessing large hardware structures typically required to buffer and process such instruction window sizes significantly degrade the cycle time. This paper proposes a novel Checkpoint Processing and Recovery (CPR) microarchitecture, and shows how to implement a large instruction window processor without requiring large structures thus permitting a high clock frequency. We fo ...

2 [Microprocessor architecture: Increasing the number of effective registers in a low-power processor using a windowed register file](#)

Rajiv A. Ravindran, Robert M. Senger, Eric D. Marsman, Ganesh S. Dasika, Matthew R. Guthaus, Scott A. Mahlke, Richard B. Brown

 October 2003 **Proceedings of the 2003 international conference on Compilers, architectures and synthesis for embedded systems**
Full text available: [pdf\(450.71 KB\)](#)
 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Low-power embedded processors utilize compact instruction encodings to achieve small code size. Instruction sizes of 8 to 16 bits are common. Such encodings place tight restrictions on the number of bits available to encode operand specifiers, and thus on the number of architected registers. The central problem with this approach is that performance and power are often sacrificed as the burden of operand supply is shifted from the register file to the memory due to the limited number of register ...

Keywords: embedded processor, graph partitioning, instruction encoding, low-power, register window, window assignment

3 [Exploiting instruction level parallelism in processors by caching scheduled groups](#)

Ravi Nair, Martin E. Hopkins

 May 1997 **ACM SIGARCH Computer Architecture News, Proceedings of the 24th annual international symposium on Computer architecture**, Volume 25 Issue 2
Full text available: [pdf\(2.01 MB\)](#)
 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern processors employ a large amount of hardware to dynamically detect parallelism in single-threaded programs and maintain the sequential semantics implied by these programs. The complexity of some of this hardware diminishes the gains due to parallelism because of longer clock period or increased pipeline latency of the machine. In this paper we propose a processor implementation which dynamically schedules groups of instructions while executing them on a fast simple engine and caches them f ...

4 Pseudo vector processor based on register-windowed superscalar pipeline

K. Nakazawa, H. Nakamura, H. Imori, S. Kawabe


December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**

Full text available:  pdf(1.19 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

5 The Clipper processor: instruction set architecture and implementation

W. Hollingsworth, H. Sachs, A. J. Smith

February 1989 **Communications of the ACM**, Volume 32 Issue 2


Full text available:  pdf(4.67 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Intergraph's CLIPPER microprocessor is a high performance, three chip module that implements a new instruction set architecture designed for convenient programmability, broad functionality, and easy future expansion.

6 Instruction-level DFT for testing processor and IP cores in system-on-a-chip

Wei-Cheng Lai, Kwang-Ting Cheng

June 2001 **Proceedings of the 38th conference on Design automation**

Full text available:  pdf(75.03 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Self-testing manufacturing defects in a system-on-a-chip (SOC) by running test programs using a programmable core has several potential benefits including, at-speed test-ing, low DFT overhead due to elimination of dedicated test circuitry and better power and thermal management during testing. However, such a self-test strategy might require a lengthy test program and might achieve a high enough fault coverage. We propose a DFT methodology to improve the fault coverage and reduce the test p ...

7 Co-synthesis of pipelined structures and instruction reordering constraints for instruction set processors

Ing-Jer Huang

January 2001 **ACM Transactions on Design Automation of Electronic Systems**

(TODAES), Volume 6 Issue 1

Full text available:  pdf(1.58 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper presents a hardware/software co-synthesis approach to pipelined ISP (instruction set processor) design. The approach synthesizes the pipeline structure from a given instruction set architecture (behavioral) specification. In addition, it generates a set of reordering constraints that guides the compiler back-end (reorderer) to properly schedule instructions so that possible pipeline hazards are avoided and throughput is improved. Co-synthesis takes place while resolving ...

Keywords: compiler instruction optimization\, instruction set processor, pipeline hazards, pipeline taxonomy, synthesis

8 Early load address resolution via register tracking

Michael Bekerman, Adi Yoaz, Freddy Gabbay, Stephan Jourdan, Maxim Kalaev, Ronny Ronen

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Additional Information:

Full text available:  [pdf\(143.17 KB\)](#)[full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Higher microprocessor frequencies accentuate the performance cost of memory accesses. This is especially noticeable in the Intel's IA32 architecture where lack of registers results in increased number of memory accesses. This paper presents novel, non-speculative technique that partially hides the increasing load-to-use latency, by allowing the early issue of load instructions. Early load address resolution relies on register tracking to safely compute the addresses of memory refere ...


9 IMPACT: an architectural framework for multiple-instruction-issue processors

Pohua P. Chang, Scott A. Mahlke, William Y. Chen, Nancy J. Warter, Wen-mei W. Hwu
April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture**, Volume 19 Issue 3

Full text available:  [pdf\(803.91 KB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

10 Register connection: a new approach to adding registers into instruction set architectures

Tokuzo Kiyohara, Scott Mahlke, William Chen, Roger Bringmann, Richard Hank, Sadun Anik, Wen-Mei Hwu
May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture**, Volume 21 Issue 2

Full text available:  [pdf\(1.07 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Code optimization and scheduling for superscalar and superpipelined processors often increase the register requirement of programs. For existing instruction sets with a small to moderate number of registers, this increased register requirement can be a factor that limits the effectiveness of the compiler. In this paper, we introduce a new architectural method for adding a set of extended registers into an architecture. Using a novel concept of connection, this method allows the data stored in ...

11 Register-sensitive selection, duplication and sequencing of instructions

Vivek Sarkar, Mauricio J. Serrano, Barbara B. Simons
June 2001 **Proceedings of the 15th international conference on Supercomputing**


Full text available:  [pdf\(235.16 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper, we present a new framework for *selecting, duplicating and sequencing* instructions so as to decrease register pressure. The motivation for this work is to target current and future high-performance processors where reductions in register pressure in the compiled programs can lead to improved performance.

For instruction selection and duplication, a unique feature of our approach is the ability to perform these transformations on intermediate-language instru ...

12 An elementary processor architecture with simultaneous instruction issuing from multiple threads

Hiroaki Hirata, Kozo Kimura, Satoshi Nagamine, Yoshiyuki Mochizuki, Akio Nishimura, Yoshimori Nakase, Teiji Nishizawa
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture**, Volume 20 Issue 2

Full text available:  [pdf\(1.03 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In this paper, we propose a multithreaded processor architecture which improves machine throughput. In our processor architecture, instructions from different threads (not a single thread) are issued simultaneously to multiple functional units, and these instructions can begin execution unless there are functional unit conflicts. This parallel execution scheme greatly improves the utilization of the functional unit. Simulation results show that by



executing two and four threads in parallel ...

13 The store-load address table and speculative register promotion

Matthew Postiff, David Greene, Trevor Mudge

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  [pdf\(170.83 KB\)](#)

 [ps\(2.97 MB\)](#) 


Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

[Publisher Site](#)

14 Multiple instruction issue in the NonStop cyclone processor

Robert W. Horst, Richard L. Harris, Robert L. Jardine

May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture**, Volume 18 Issue 3

Full text available:  [pdf\(1.06 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the architecture for issuing multiple instructions per clock in the NonStop Cyclone Processor. Pairs of instructions are fetched and decoded by a dual two-stage prefetch pipeline and passed to a dual six-stage pipeline for execution. Dynamic branch prediction is used to reduce branch penalties. A unique microcode routine for each pair is stored in the large duplexed control store. The microcode controls parallel data paths optimized for executing the most frequent instr ...

15 IMPACT: an architectural framework for multiple-instruction-issue processors

Pohua P. Chang, Scott A. Mahlke, William Y. Chen, Nancy J. Warter, Wen-mei W. Hwu

August 1998 **25 years of the international symposia on Computer architecture (selected papers)**


Full text available:  [pdf\(910.04 KB\)](#)

Additional Information: [full citation](#), [references](#), [index terms](#)

16 Memory interfacing and instruction specification for reconfigurable processors

Jeffrey A. Jacob, Paul Chow

February 1999 **Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays**

Full text available:  [pdf\(1.77 MB\)](#)


Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: FPGA, memory coherence, memory interfacing, reconfigurable computer, reconfigurable processor

17 Register traffic analysis for streamlining inter-operation communication in fine-grain parallel processors

Manoj Franklin, Gurindar S. Sohi

December 1992 **ACM SIGMICRO Newsletter , Proceedings of the 25th annual international symposium on Microarchitecture**, Volume 23 Issue 1-2

Full text available:  [pdf\(1.31 MB\)](#)


Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

18 Two-level hierarchical register file organization for VLIW processors

Javier Zalamea, Josep Llosa, Eduard Ayguadé, Mateo Valero

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  [pdf\(154.90 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)


 [ps\(843.85 KB\)](#)

 [Publisher Site](#)

19 [Extending the power of short-wordlength processors by means of context-dependent machine instructions](#)

C K Yuen

October 1981 **ACM SIGARCH Computer Architecture News**, Volume 9 Issue 6

Full text available:  [pdf\(669.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

It is shown that the concept of context-dependent machine instructions may be used in the architectural design of processors with short wordlengths, such as 8-bit microprocessors, in order to increase the capabilities of such machines above those of currently available models.

Keywords: computer architecture, context, machine instructions, microprocessor, processor status, register references

20 [Code selection for media processors with SIMD instructions](#)

Rainer Leupers

January 2000 **Proceedings of the conference on Design, automation and test in Europe**

Full text available:  [pdf\(147.30 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)





 [Publisher Site](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)